Document Clustering in Reduced Dimension Vector Space

Kristina Lerman
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
Email: lerman@isi.edu

Abstract

Document clustering is a popular tool for automatically organizing a large collection of texts. Clustering algorithms are usually applied to documents represented as vectors in a high dimensional term space. We investigate the use of Latent Semantic Analysis to create a new vector space, that is the optimal representation of the document collection. Documents are projected onto a small subspace of this vector space and clustered. We compare the performance of clustering algorithms when applied to documents represented in the full term space and in reduced dimension subspace of the LSA-generated vector space. We report significant improvements in cluster quality for LSA subspaces with optimal dimensionality. We discuss the procedure for determining the right number of dimensions for the subspace. Moreover, when this number is small, the total running time of the clustering algorithm is comparable to the one that uses the full term space.

Introduction

Clustering is used to partition a set of data so objects in the same cluster are more similar to one another than they are to objects in other clusters. In the field of information retrieval (IR), document clustering is used to automatically organize large collection of retrieval results, grouping together documents that belongs to the same topic in order to facilitate user's browsing of retrieval results [7]. It is also used in word sense disambiguation, as well as many other applications.

Most clustering algorithms use the vector space model of IR [11], in which text documents are represented as a set of points in a high dimensional vector space. Each direction of the vector space corresponds to a unique term in the document collection and the component of a document vector along a given direction corresponds to the importance of that term to the document. Similarity between two texts is traditionally measured by the cosine of the angle between their vectors, though Cartesian distance is also used. Documents judged to be similar by this measure are grouped together by the clustering algorithm. These algorithms are notoriously slow, because the time required to compute a similarity score for each pair of documents is proportional to the size of the smaller document. Different methods for reducing the dimensionality of the vector space, thereby reducing the complexity of document representation and speeding up similarity computation times, have been investigated [12]. One method involves keeping only N most important terms from each document (as judged by the chosen term weighting scheme), where N is much smaller than document size. Another method applies Latent Semantic Analysis (LSA) to the document collection to create a new abstract vector space, which has a special property that vectors describing this vector space are ordered according to their importance to describing the document collection. Documents are projected onto a small subspace of this vector space and clustered. When an optimal number of dimensions of the subspace is picked, we observe significant improvements in clustering quality as compared with results produced by applying the same algorithms to documents projected onto the full term space. We discuss LSA and its applications to text analysis, and present a method for picking the number of dimensions for the LSA subspace. For our test collection, the substantial speed up times of the clustering in reduced dimension vector space offset the additional costs associated with computing LSA vectors, and we gain improved clustering quality without sacrificing performance.

Background

Latent Semantic Analysis has been proposed [4, 2] as a tool for uncovering common patterns of word usage across a large number of documents. It is similar to the Principal Component Analysis long used by

statisticians to analyze data sets composed of many highly correlated variables and to represent them in terms of a few uncorrelated variables [1]. LSA uses a mathematical technique called Singular Value Decomposition (SVD) to create a new abstract vector space that is the best representation of the document collection in the least-squares sense. SVD also computes two sets of orthogonal (independent) vectors that form a basis of this vector space. One set of vectors provides a transformation from the original vector space of terms to a new LSA-space, and the other set provides a transformation from the document space to the new LSA-space. Moreover, SVD returns an ordering of these vectors, so that the first vector is the most informative, that is it describes the strongest regularities of word usage in the document collection. The second vector describes those aspects of word usage not captured by the first vector, and so on to the last and least informative vector. In addition to simplifying the description of the document collection, LSA can yield valuable insights into the relationships among original terms. It has been used in IR applications to retrieve relevant documents that do not share any terms with the query [4]. LSA has also been proposed as a model for children's learning of new words [8], as well as for measuring textual coherence within and across documents [6], and other applications [14].

Singular value decomposition is used to rewrite an arbitrary rectangular matrix, such as a term-document matrix, as a product of three other matrices – a matrix of left singular vectors, a diagonal matrix of singular values, and a matrix of right singular vectors. The left singular vectors provide a mapping from the term space to a newly generated abstract vector space, while the right singular vectors provide a mapping from the document space to the new space. The elements of the diagonal matrix, the singular values, appear in order of decreasing magnitude. One of the more important theorems of SVD states that a matrix formed from the first k singular triplets of the SVD (left vector, singular value, right vector combination) is the best approximation to the original matrix that uses k degrees of freedom. The technique of approximating a data set with another one having fewer degrees of freedom is known as dimensional reduction, or noise filtering. It works because the leading singular triplets capture the strongest, most meaningful, regularities in data – in LSA these are patterns of word usage in the document collection. The later triplets represent less important, possibly spurious, patterns. Ignoring them actually improves analysis, though there is the danger that by keeping too few degrees of freedom, or dimensions of the abstract vector space, some of the important patterns will be lost [9]. To pick an optimal number of degrees of freedom for the approximation, we borrow techniques from Principal Component Analysis [1]. We plot the singular values in order and look for a break or a discontinuity in the slope, i.e., the point to the left of which the singular values are decreasing much faster than to the right. Though the time complexity of the SVD algorithm is cubic in the number of triplets calculated, dimensionality reduction using the SVD is an efficient process, because the calculation of each singular triplet only depends on the values of the preceding triplets.

The first k singular vectors describe a k-dimensional subspace of the abstract LSA vector space. The left singular vectors project terms onto the k-dimensional subspace, and can have positive or negative components. The negative components represent terms that are anti-correlated with the positive components. In other words, for documents in which positively weighted terms tend to appear, the negatively weighted terms tend *not* to appear. The right singular vectors project documents onto the k-dimensional subspace. Documents, now represented as k-dimensional vectors, are clustered using a standard clustering algorithm, such as the hierarchical agglomerative clustering algorithm known as SLINK [10].

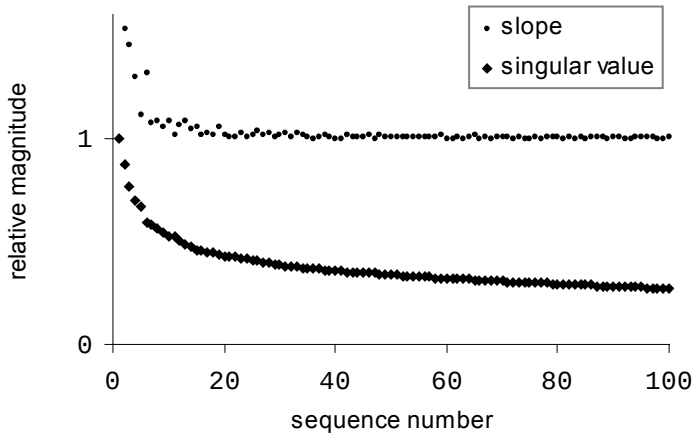
Results

The collection we chose to use for validation was composed of 1000 documents – 200 documents from each of the five TREC topics (125, 158, 163, 183, 191). The text of every document in the collection was tokenized, all tags and punctuation marks were stripped from the text, characters were converted to lower case. Stop words and terms beginning with numbers were ignored, while the occurrences of the remaining (unstemmed) terms were counted. A term-document matrix, whose elements a_{ij} give the weight of term i in document j, was constructed from the unique terms. We used a somewhat modified form of the traditional TFIDF term weighting scheme [5],

$$a_{ij} = \frac{\log t_{ij}}{n_j} \cdot \log\left(\frac{n}{d_i}\right),$$

where t_{ij} is the occurrence count of term i in document j, n_j is the length of document j, d_i is the number of documents in which i appears, and n is the total number of documents in the collection. We used the same weighting to represent documents in term space, that is the component i of document vector j is given by a_{ij}.

The single-vector Lanczos method from SVDPACKC [3] was used to decompose the term-document matrix into singular triplets. Though the computational cost of a full SVD of very large matrices can be prohibitive, it took on the order of minutes to compute the SVD of a matrix of a thousand documents of 20Kb average size on a Pentium 400MHz processor computer. Computation times are drastically reduced



when only a small number of triplets is calculated (see Table 2).

Figure 1 – Singular values of the SVD of the term-document matrix of 1000 documents from five TREC topics. The bottom curve shows the relative magnitudes of the first 100 singular values plotted in order. The top curve shows the absolute value of the slope of the bottom curve, which indicates the rate at which singular values are decreasing. The value of slope are magnified by a factor of ten, and displaced from the origin by 1.0 for emphasis.

SVD calculates at most 1000 singular triplets. Relative magnitudes of the first 100 singular values are plotted in order in the lower portion of Figure 1. To help identify discontinuity in the slope of the singular value curve, we plot in the top part of the figure the difference between each successive pair of singular values, magnified by factor of 10 and displaced by 1 from the origin for emphasis. There is a marked slope discontinuity around the 6th singular value. To the right of the discontinuity the singular values change much more gradually than to the left. We conclude that the first six singular triplets capture the most important dimensions of meaning in the document collection, while the higher triplets account for patterns of an ever diminishing importance. While it appears that significantly erring on the right side of the true discontinuity does not degrade clustering results, significantly erring on the left side can lead to diminished performance (see Table 1).

Documents were projected onto several subspaces of different dimensionality, and clustered using the SLINK algorithm. SLINK organizes documents in a tree-like structure. Our clustering algorithm descended the tree, forming clusters from documents in a subtree whose size falls below a specified cutoff value. Clusters with fewer than ten documents were ignored. For comparison, documents were also clustered in the original term space using the same procedure. When the LSA subspace has dimensionality close to the

optimal value of six, the clusters reproduce the original topic divisions of the test collection. Even for the greater numbers of dimension, say 50, the clusters were bigger and purer than they were for documents clustered in term space. To quantify this observation we define precision per cluster as the ratio of the number of documents correctly assigned to the cluster to the size of the cluster. We define recall per cluster as the ratio of the number of documents correctly assigned to the cluster to the "ideal" size of the cluster, which was 200 and 100 in our experiments, the value of the cutoff size used with the SLINK algorithm. Recall and precision values were averaged over all clusters of size ten and above for each subspace, and the results are listed in Table 1. For $k = 5$, $k = 6$ and $k = 10$ the precision and recall values are both high, consistent with the observation that original topic divisions are recovered by the clustering procedure when the dimension of the abstract vector space is close to its optimal value. As the dimensionality of the subspace was increased, fewer documents were assigned to clusters, and clusters become less pure. Results for $k = 500$ are similar to those achieved by clustering in term space. When not enough dimensions were picked to describe the document collection, as in the $k = 3$ case, clustering performance was degraded substantially. For SLINK100, the clusters are more likely to be smaller, because of the way they are formed from the tree, but also more likely to be purer. The smaller cluster sizes lead to smaller average recall per cluster. We still see significant improvement in quality when documents were clustered in the reduced dimension vector space. It is worth mentioning that the results below are similar to those obtained by clustering a collection composed only of documents relevant to the TREC topics.

subspace	recall	precision	#total	#clusters
		SLINK200		
$k = 3$	0.275	0.853	851	14
$k = 5$	0.963	0.991	972	5
$k = 6$	0.978	0.997	981	5
$k = 10$	0.591	0.998	948	8
$k = 50$	0.367	0.978	824	11
$k = 100$	0.232	0.923	758	14
$k = 500$	0.155	0.774	487	10
term space	0.152	0.813	574	12
		SLINK100		
$k = 3$	0.336	0.891	698	19
$k = 5$	0.538	0.981	649	12
$k = 6$	0.556	1.000	612	11
$k = 10$	0.512	1.000	666	13
$k = 50$	0.368	0.985	672	18
$k = 100$	0.336	0.965	621	18
$k = 500$	0.316	0.845	473	12
term space	0.300	0.873	562	15

Table 1 – Results of clustering 1000 documents in different vector spaces using SLINK algorithm with two cutoffs for forming clusters: 200 and 100. The first column indicates the dimensionality of the abstract subspace generated by the SVD, or whether the documents were clustered in term space. Recall and precision per cluster values were calculated for every cluster of size ten and above, and averaged over all clusters in each vector space. Total number of documents assigned to clusters, and the number of clusters, are displayed in the last two columns.

subspace	time (s)	subspace	time (s)
$k = 3$	33	$k = 50$	42
$k = 5$	33	$k = 100$	53
$k = 6$	33	$k = 500$	355
$k = 10$	34	term space	30

Table 2 – Time, in seconds, taken to cluster 1000 documents on a 400 MHz Pentium personal computer. The times include calculation of singular triplets, where appropriate.

Running times of the entire clustering procedure on a 400 MHz Pentium processor PC, including SVD calculations where appropriate, are listed in Table 2. When the document collection is best described by few degrees of freedom, as is true for our test collection, projection onto the SVD-generated subspace does not appear to add significant overhead to the run times of the clustering algorithm, while the clustering results are significantly improved.

Schuetze *et al*. [12], report using different projection methods for clustering 74000 documents from 49 TREC topics. They compare projections onto LSA subspaces of dimensions 20 (LSA-20), 50 (LSA-50) and 150 (LSA-150) with clustering in term space when each document is represented by its 20 (TF20) and 50 (TF50) most weighty terms. They do not find degradations in clustering quality using these projection methods when compared to clustering in full term space, except for the radical truncation TF20. However, they do not report any significant improvement for LSA subspace clustering, or significant difference between LSA subspace clustering and TF50. When we run our clustering algorithms on truncated term space, we find average per cluster recall (precision) values of 0.227(0.862) for TF20 and 0.154(0.812) for TF50. These values are comparable to full term space results, but they are still much worse than $k = 6$ results. The difference in our findings might stem from the fact that we are using a smaller test collection and a different evaluation methodology. It might also originate from clustering in non-optimal LSA subspace. It would be interesting to find the optimal number of dimensions that describe their larger test collection by the methods described above.

Discussion

It is advantageous to cluster documents in a reduced dimension abstract LSA space rather than term space, because original term space is not a good representation of the document collection as a whole. Noise tends to obscure regularities in the data. LSA, through the SVD algorithm, finds a better representation of the collection, a representation in which structures, such as clusters, become apparent. A closer look at this representation uncovered by the SVD suggests why clustering works well in LSA subspace. Below is a list of the five TREC topics which make up our test collection, followed by the list of the ten largest magnitude positive and negative components of the first seven left singular vectors computed by SVD.

Topic 125: Anti-smoking Actions by Government
Topic 158: Term limitations for members of the U.S. Congress
Topic 163: Vietnam Veterans and Agent Orange
Topic 183: Asbestos Related Lawsuits
Topic 191: Efforts to Improve U.S. Schooling

1. Manville[0.453] asbestos[0.447] bankruptcy[0.255] trust[0.202] company[0.179] court[0.157] plan[0.138] veterans[0.133] claims[0.130] reorganization[0.125]

2. manville[0.243] asbestos[0.142] bankruptcy[0.127] trust[0.102] reorganization[0.066] plan[0.058] company[0.052] insurance[0.043] stock[0.038] macarthurr[0.037]
veterans[-0.492] vietnam[-0.406] orange[-0.241] agent[-0.239] study[-0.119] exposure[-0.115] readjustment[-0.115] meeting[-0.103] war[-0.093] committee[-0.085]

3. smoking[0.370] tobacco[0.320] school[0.173] education[0.167] smokers[0.144] cigarette[0.139] students[0.130] teachers[0.127] schools[0.120] advertising[0.116]
veterans[-0.218] manville[-0.204] vietnam[-0.186] orange[-0.116] agent[-0.116] bankruptcy[-0.099] trust[-0.093] reorganization[-0.056] readjustment[-0.050] tcdd[-0.046]

4. smoking[0.289] tobacco[0.264] cigarette[0.116] advertising[0.101] smokers[0.101] cigarettes[0.098] ban[0.076] smoke[0.072] industry[0.072] anti[0.069]
education[-0.320] school[-0.310] teachers[-0.259] students[-0.249] schools[-0.225] teacher[-0.153] percent[-0.114] high[-0.095] average[-0.091] bush[-0.083]

5. manville[0.399] trust[0.186] bankruptcy[0.172] smoking[0.163] tobacco[0.141] reorganization[0.110] plan[0.105] smokers[0.063] macarthur[0.0589] advertising[0.057]
asbestos[-0.501] raymark[-0.214] concentrations[-0.207] fibreboard[-0.179] lung[-0.142] fibers[-0.132] epa[-0.114] concentration[-0.097] raytech[-0.095] bodies[-0.088]

6. smoking[0.140] concentrations[0.111] tobacco[0.110] school[0.094] manville[0.090] education[0.082] students[0.073] lung[0.071] teachers[0.070] smokers[0.070]
term[-0.282] limits[-0.278] congress[-0.227] fibreboard[-0.223] voters[-0.172] limit[-0.165] house[-0.151] court[-0.139] senate[-0.136] incumbents[-0.128]

7. concentrations[0.287] asbestos[0.154] term[0.144] limits[0.143] concentration[0.137] bodies[0.137] fibers[0.129] lung[0.128] manville[0.119] congress[0.108]
fibreboard[-0.475] pacific[-0.236] indemnity[-0.148] raymark[-0.121] claims[-0.120] louisiana[-0.119] company[-0.117] insurers[-0.097] related[-0.093] insurance[-0.091]

Each vector consists of terms that are correlated in some way across documents in the test collection. The value in square brackets is the strength of the term in the vector, and it measures the degree of correlation between the term and the word pattern expressed by the singular vector. As we have mentioned above, the correlations can be positive and negative, the latter measuring anti-correlation between terms. Though it might be tempting to assign singular vectors to individuals topics, it is a mistake to do so. The third vector, for example, seems to be a mixture of words from topics 125 and 191. Also, the first five vectors do not correspond to the five topics – it is not until the sixth vector that we see strong contributions from terms related to topic 158. Taken together, the first six singular vectors capture the main degrees of freedom of the document collection, namely those described by the five topics. This is the reason that projecting documents onto the $k = 6$ subspace of the abstract LSA space seems to produce the best clustering results. When many fewer than six degrees of freedom are used to describe the test collection, for example $k = 3$, the representation is not sufficiently rich, and many documents are misclassified as a result.

In the future, we would like to extend these methods to larger document collections, and to collections that do not have well defined topics, such as documents returned by Internet search engines in response to a query. For instance, we have done some work on clustering documents that are returned by search engines in response to queries containing ambiguous terms. We hypothesize that given a large number of documents, the first few singular triplets calculated by the SVD will correspond to different senses of the ambiguous term. Clustering using those degrees of freedom will separate documents into groups that share the same word sense of the query term. While the preliminary results look encouraging, we find that a typical collection of 500 Web documents contains many more distinct topics that the number of different senses of the query term might indicate. We hope that larger collections of documents will have better word pattern statistics.

Acknowledgments

I would like to thank Ed Hovy, Richard Ross and Chin-Yew Lin for many helpful discussions. This work was supported by

References

1. Afifi, A. A., & Clark, V. (1996) *Computer Aided Multivariate Analysis, 3rd ed.* London: Chapman & Hall.

2. Berry, M. W., Dumais, S. T., & O'Brien, G. W. (1995). Using linear algebra for intelligent information retrieval. *SIAM: Review, 37*, 573—595.

3. Berry, M. W., Do, T., O'Brien, G. W., Krishna, V., & Varadhan, S. (1996) SVDPACKC (Version 1.0) and User's Guide.

4. Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science, 41,* 391—407.

5. Dumais, S. T. (1995) Latent semantic indexing (LSI): TREC-3 Report. In D. Harman (ed.), *The Third Text Retrieval Conference (TREC3)* National Institute of Standards and Technology Special Publication 500-236.

6. Foltz, P., Kintsch, W., & Landauer, T. K. (1998). Measurement of text coherence with latent semantic analysis. *Discourse Processes, 25*, 285—307.

7. M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis. In *Proceedings of SIGIR '96*, pp. 76—84, 1996.

8. Landauer, T. K. & Dumais, S. T. (1997) A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, *104*, 211—240.

9. Landauer, T. K., Foltz, P. W., & Laham, D. (1998) An introduction to latent semantic analysis. *Discourse Processes, 25*, 259—284.

10. Rasmussen, E. (1992) Clustering algorithms. In W. F. Frakes and R. Baeza-Yates (Eds.) *Information Retrieval: Data Structures and Algorithms* (pp. 419-442). New Jersey: Prentice Hall.

11. Salton, G., & McGill, M. (1983) *Introduction to Modern Information Retrieval.*New York: McGraw—Hill.

12. H. Schutze and C. Silverstein. (1997) Projections for efficient document clustering. In *Proceedings of SIGIR '97*, pp. 74—81, 1997.

13. H. Shutze (1998), Automatic Word Sense Discrimination. *Computational Linguistics, 24,* pp. 97—123.

14. Wolfe, M. B. W., Shreiner, M. E., Rehder, R., Laham, D, Foltz, P., Kintsch, W., & Landauer, T. K. (1998). Learning from text: matching readers and texts by latent semantic analysis. *Discourse Processes, 25*, pp. 309—336.